In [1]:

```
%matplotlib inline
```

In [2]:

```
# uNMR COMS is the first code operating the HMC832 PLL on the uNMR Board

# Set-Up
# 1. Connect the FTDI USB-RS485-WE to the uNMR board
# 2. Connect 5.2 - 5.5Vdc to the uNMR board
# 3. Programme the TMS320F28335PTPQ with the riquired code
# 4. Click "Cell" above and then click "Run All"
# 5. Measure PLL output initial value is
```

In [3]:

```
#%matplotlib inline                                # Polts plots in scrip

import time                                        # Timer units seconds
import math                                        # math functions
import numpy as np                                 # multi-dimensional arrays/
 matrices
import pandas as pd                                # building block for data a
nalysis
import minimalmodbus                               # modbus RTU serial comms m
odule
import matplotlib.pyplot as plt                    # Python plotting library

instrument = minimalmodbus.Instrument('com3', 1)   # port name, slave address
 (in decimal)
instrument.serial.baudrate = 57600                 # Baud
instrument.serial.bytesize = 8
instrument.serial.parity   = 'E'                   # Parity

instrument.serial.stopbits = 1
instrument.serial.timeout  = 0.05                  # seconds

instrument.mode = minimalmodbus.MODE_RTU           # rtu or ascii mode
```

In [4]:

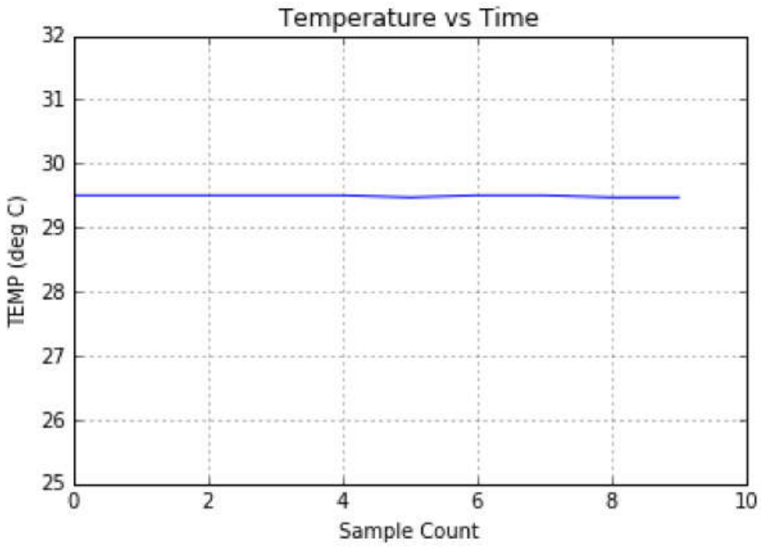
```
vtune = instrument.read_registers(6,1)             # read case 6
print (vtune)
```

```
[3074]
```

In [7]:

```
# Temperature sensor, board SPI, routine
# This routine requires "import numpy as np" for np.linspace
# This routine requires  "import time"


temp_read = []                                      # array temp_read

count_max = 10                                    # sample count max
count_min = 0                                       # sample count min
yaxis_min = 25                                      # yaxis min
yaxis_max = 32                                      # yaxis max
print 'Count  Temp (degC)'                          # Column header

for i in range(count_max):
    print '{:02f}'.format(i)                       # Prints every result
    time.sleep(.2)                                  # loop delay 0.2s
    val = instrument.read_register (7, 1)           # read modbus reg 7 decima
l 1
    val2 = val * 0.3125                             # raw data to deg C conver
sion
    print ' ', i, '  ', val2                       # prints column header
    temp_read += [val2]

plt.plot( temp_read )                              # Plot results
plt.title('Temperature vs Time')
plt.xlabel('Sample Count')
plt.ylabel('TEMP (deg C)')
plt.grid( True )

plt.axis([count_min, count_max, yaxis_min, yaxis_max])
plt.show()

#print temp_read
```

```
Count  Temp (degC)
0.000000
  0    29.5
1.000000
  1    29.5
2.000000
  2    29.5
3.000000
  3    29.5
4.000000
  4    29.5
5.000000
  5    29.46875
6.000000
  6    29.5
7.000000
  7    29.5
8.000000
  8    29.46875
9.000000
  9    29.46875
```



```
In [ ]:
```